ENHANCEMENTS OF SPARSE CLUSTERING WITH RESAMPLING AND CONSIDERATIONS ON TUNING PARAMETER

by

Wenzhu Bi

B.E., Shanghai Jiao Tong University, Shanghai, China, 2000

M.S., Duquesne University, 2004

Submitted to the Graduate Faculty of

the Department of Biostatistics

Graduate School of Public Health in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2012

UNIVERSITY OF PITTSBURGH

GRADUATE SCHOOL OF PUBLIC HEALTH

This dissertation was presented

by

Wenzhu Bi

It was defended on

April 9th 2012

and approved by

Lisa A. Weissfeld PhD, Professor, Department of Biostatistics

Graduate School of Public Health, University of Pittsburgh

George C. Tseng ScD, Associate Professor, Department of Biostatistics

Graduate School of Public Health, University of Pittsburgh

Yan Lin PhD, Research Assistant Professor, Department of Biostatistics

Graduate School of Public Health, University of Pittsburgh

Julie C. Price PhD, Professor, Department of Radiology

School of Medicine, University of Pittsburgh

Dissertation Director: Lisa A. Weissfeld PhD, Professor, Department of Biostatistics

Graduate School of Public Health, University of Pittsburgh

Copyright © by Wenzhu Bi
2012

ENHANCEMENTS OF SPARSE CLUSTERING WITH RESAMPLING AND CONSIDERATIONS ON TUNING PARAMETER

Wenzhu Bi, PhD

University of Pittsburgh, 2012

Clustering methods are widely used to explore subgroupings in data when the true group membership is unknown. These techniques are very useful when identifying potential subpopulations of interest in the medical and public health setting. Examples of these types of subpopulations include subjects who have certain gene expression profiles related to a cancer subtype, and subjects who are in the very early, asymptomatic phase, of a chronic illness. All of these examples are of great public health relevance.

Many of the datasets of interest arise from the development of new technologies and are subject to the common problem where p, the number of variables, is significantly larger than the sample size, n. The relatively small sample size, n, may result from the difficulties of subject recruitment and/or the financial burden of the actual data collection in fields such as imaging and genetic analysis. The earlier approaches to clustering treat all of the variables equally, which may not work well when not all of them are relevant to the subgroupings. Clustering methods with variable selection, also called sparse clustering, have been recently developed to deal with this problem. We propose a method to add resampling onto sparse clustering to improve upon the current clustering methodology. The addition of resampling methods to sparse clustering results in variable selection that is more accurate. The method is also used to assign an"observed proportion of cluster membership" to each observation, providing a new metric by which to measure membership certainty. The performance of the method is studied via simulation and illustrated in the motivating data example.

We also propose an alternative approach for the choice of tuning parameter based on

an adjusted Bayesian Information Criterion (BIC). Variable selection in sparse clustering is realized by applying Lasso or related penalties and the tuning parameter for these penalties has to be determined beforehand. The gap statistic, a distance-based approach, is used to choose the tuning parameter through permutation and it may behave poorly at times. The proposed BIC approach is an alternative developed under the more sophisticated model-based likelihood framework. Its performance is evaluated with simulations.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PREFACE

I thank my primary advisor Dr. Lisa A. Weissfeld for her vision, guidance, and help. She introduced me to the field of brain imaging research. With her encouragement and support, I have gained invaluable training and experiences. She supported me to join the graduate training program at the Center for the Neural Basis of Cognition to gain in-depth training in neuroscience. She encouraged me to take courses and to attend imaging workshops throughout the country to build up the skill set. During the dissertation work, she allowed me to pursue topics of my own interest and shared her wisdom along the way.

I thank my committee member and also co-advisor, Dr. George C. Tseng, for his invaluable advice. He inspired my interest and confidence in clustering through classroom teaching and individual research meetings. He provided critical feedback for many aspects of the work. His research group provided me crucial support for using the Linux computing server. I also thank my committee member Dr. Yan Lin for her insightful suggestions to improve the quality of this dissertation.

Special gratitude goes to my imaging mentor Dr. Julie C. Price. She gave me the opportunity to work on the radiotracer Pittsburgh compound-B, a major scientific breakthrough in Alzheimer's disease research. Like Dr. Weissfeld, Dr. Price has been very supportive in every single step of my career development. I am thankful for her encouragement, support, and guidance.

Hearty thanks goes to my classmates, colleagues and friends in Pittsburgh for their help and support. I sincerely thank my closest friends and colleagues Dr. Bedda L. Rosario and Rhaven L. Coleman for their encouragements and support. At last, I thank my husband Darrick W. Mowrey and our family for their love, support and encouragements.

1.0 INTRODUCTION

Large datasets have been easily generated by the advancement in computation in the last twenty years. They exist in the fields of genetics, imaging, chemometrics and many others. Many of these datasets have the property of "high-dimension, low sample size" ($p \gg n$), which means that the number of variables, p, is much larger than the number of observations, n. For example in DNA microarray data, the expression levels for tens of thousands genes can be observed while there are usually dozens or hundreds of subjects in a study. In neuroimaging, there could be hundreds of thousands of voxels per brain scan with only a few dozen subjects in a study. The challenge is to extract useful information from this abundance of data and statistical learning methods have evolved to address these issues.

Statistical learning can be roughly categorized into supervised learning and unsupervised learning depending on whether the true group membership is known or not. If the true membership is known, it can serve as a guide for the construction of the statistical learning method, hence it is named supervised learning. Classification is a classic example. When the true group membership is unknown, we lose this crucial information for guidance; hence it is called unsupervised learning. Clustering is a classic example of unsupervised learning and it is also the focus of this dissertation work.

Clustering analysis usually aims to group a collection of objects into clusters "such that those within each cluster are more closely related to one another than objects assigned to different clusters" as defined by Hastie, Tibshirani and Friedman[15]. Correspondingly, the algorithms try to maximize the between-cluster difference while minimizing the within-cluster difference. The difference is called *dissimilarity*, which can be Euclidean distance (i.e., squared difference), Manhattan distance (i.e., absolute difference) or other metrics depending on the problem. Euclidean distance is most commonly used for the *dissimilarity* measure.

Typical clustering methods include k-means clustering, k-medoids clustering, hierarchical clustering and model-based clustering.

K-means clustering is one of the most popular clustering methods[15]. It and its variants have been widely applied in various fields such as computer science, engineering, business, economics, biology, chemistry, psychology, neurology, medical imaging, pharmacology and psychiatry as shown in Figure **??**. It can be traced back to work done by Lloyd in 1957[20], by Forgy in 1965[6], by MacQueen in 1967[21] and by Hartigan and Wong in 1979[13]. The algorithm was significantly improved by Hartigan and Wong(1979)[13]. K-medoids clustering is very similar to k-means. The only difference is that the cluster center is the object closest to the cluster mean while in k-means the cluster center is the cluster mean; in other words, the cluster centers are medoids rather than centroids. K-medoids clustering is more robust to outliers for this reason[18]. K-means and k-medoids require the user to specify the number of clusters while hierarchical clustering does not require so. Hierarchical clustering[16] provides a dendrogram as the results, which shows a summary of the data. The dendrogram can be built with either an *agglomerative*(bottom-up) or a *divisive*(top-down) approach. The distance between clusters can be defined in various ways such as defined by the closest pair of objects between two clusters, or the furtherest pair, or as the average of all the pairs.

K-means, k-medoids and hierarchical clustering methods are considered to be heuristic distance-based clustering methods. Parametric models have also been developed for clustering, resulting in model-based clustering techniques. The advantage of model-based clustering is the ability to draw statistical inference. A very successful model is the finite mixture model such as work done by McLachlan and Basford in 1988[25], Banfield and Raftery in 1993[1], and Fraley and Raftery in 1998[7] and 2002[8]. The last two papers discussed different covariance structures. The estimation was realized by EM algorithm and the model selection by Bayesian Information Criterion (BIC).

For a dataset, $X_{n\times p}$, with n observations and p variables per observation, earlier approaches to clustering tend to treat all of the variables equally. It may not work well when not all of them are relevant to the subgroupings, which is especially true in the case of the "high-dimension, low sample size" ($p \gg n$) problem. Clustering methods with variable selection, also called sparse clustering methods, have been recently developed to deal with this

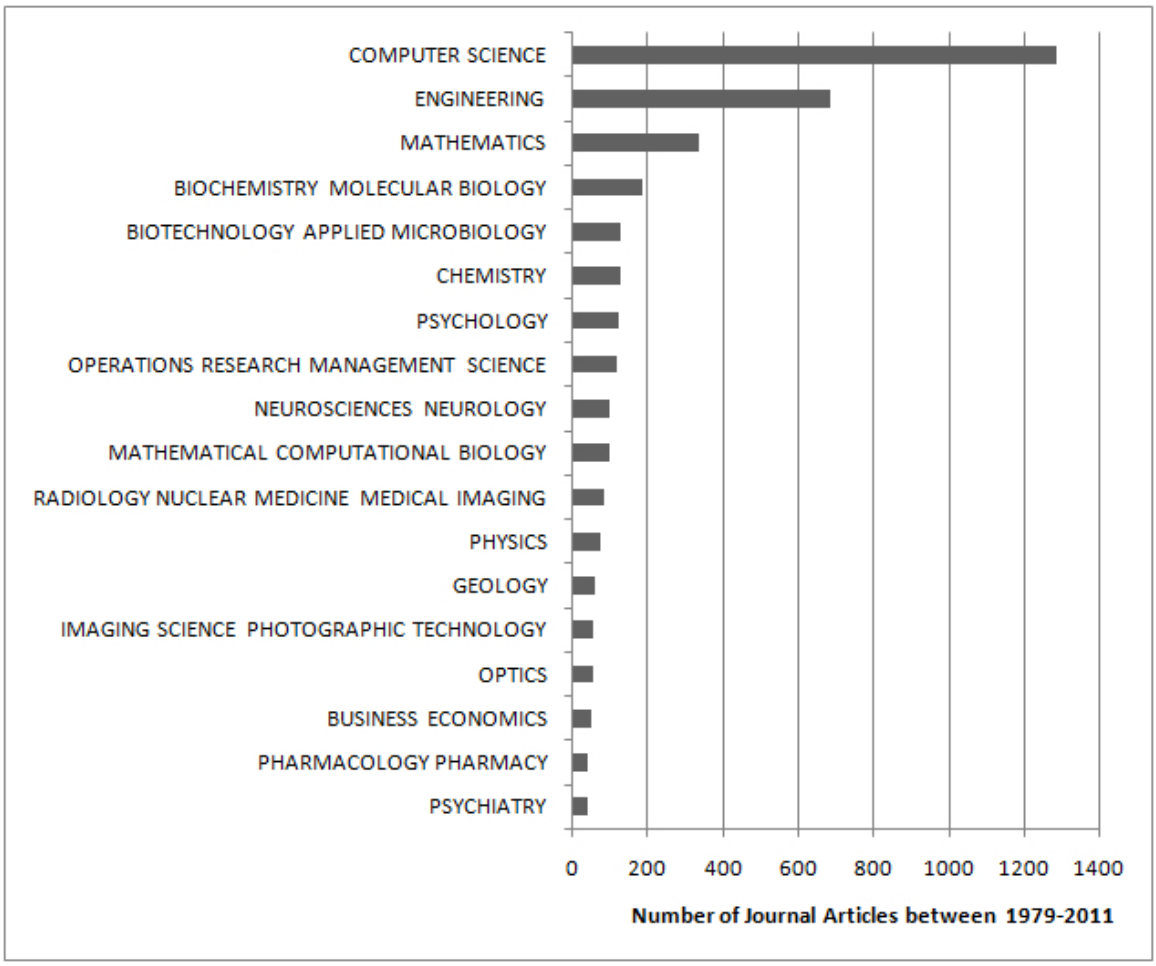


Figure 1: The number of journal articles between 1979 and 2010 with k-means clustering categorized by subject area. Shown above is a sample of the subject areas. Citation data is obtained from web of knowledge (http://wokinfo.com/).

problem. Many of the sparse clustering methods make use of an L_1/Lasso penalty or related penalties to realize variable selection. A general framework for clustering was proposed by Witten and Tibshirani in 2010 [35]. It applied both an $L1$ and an $L2$ restraint to realize variable selection. It was demonstrated with sparse k-means and sparse hierarchical clustering. These penalties were also applied to the model-based clustering framework. Pan and Shen in 2007[27] proposed a penalized model-based clustering method, in which BIC was used for model selection. The penalized mixture model assumes independence between variables and the same diagonal covariance matrix across clusters. The development of sparse clustering analysis is moving clustering into a new era and allowing us to sift the data more closely than we have ever been able to.

In this dissertation work we develop two new methods under the scope of sparse clustering. The first method is motivated by the problem of small sample size. For example, a neuroimaging dataset with a few dozens of subjects is common due to the challenges in subject recruitment and the cost of scans. Clustering of such data is important in exploring the patterns associated with different underlying biological characteristics and in searching for a disease detection threshold. Often times, due to the limitations of the sample size, one would consider issues such as "How representative is the sample of the population?" or "If we were going to construct clusters, how much confidence can we have in the clustering results?" We propose a method to add resampling onto sparse clustering to address these concerns. This method will alleviate the effects of noise and outliers on the variable selection and clustering results and generate confidence levels for the clustering results.

The second method is about the choice of the tuning parameter. The results of the *Lasso*-related penalties depend on the tuning/penalization parameter choice. The magnitude of the tuning parameter relates to the number of variables selected for clustering. The choice of the tuning parameter has been studied in regression and classification with cross-validation being the main approach. This method can not be easily applied to clustering methods due to the lack of information on the true group membership. Pan and Shen in 2007[27] used a modified BIC to determine both the number of clusters and the tuning parameter. Witten and Tibshirani in 2010[35] proposed using the gap statistic. Yet both of these methods rely on the correct specification of the tuning parameter pool. We have observed that given

a suboptimal pool the gap statistic method could choose a tuning parameter which yields poor performance when clustering, i.e. a high Classification Error Rate (CER). When this happens it usually chooses a tuning parameter that is either too small or too large, hence too few, or too many, variables are chosen for clustering. This is more likely to occur when the overlap between the clusters is more substantial. In this work, we would like to draw attention to this phenomenon and also to propose using the BIC for the choice of the tuning parameter for the sparse k-means method proposed by Witten and Tibshirani in 2010 [35].

The two methods will be described separately in the next two chapters. Chapter 2 includes an introduction to the background, the methods section, simulation study results and an application to the motivating imaging dataset. In chapter 3, we give an introduction to the background, the existing methods, scenarios where gap statistic has failed, and the proposed method to address the issue. The method is demonstrated via a real example and simulation studies. The last chapter summarizes the conclusions of this work.

2.0 SPARSE CLUSTERING WITH RESAMPLING

2.1 INTRODUCTION

Large datasets have become commonplace due to the advancement in computation and other technologies in the last twenty years. They exist in the fields of genetics, imaging, chemometrics and many others. Many of these datasets have the property of "high-dimension, small sample size" ($p \gg n$), where the number of variables, p, is much larger than the number of observations, n. For example in DNA microarray data, the expression levels for tens of thousands of genes can be observed while there are usually dozens or hundreds of subjects in a study. In neuroimaging, there could be hundreds of thousands of voxels per brain scan with only a few dozen subjects in a study. The challenge is to extract useful information from this abundance of data and statistical learning methods, including clustering, have evolved to address these issues.

Clustering aims to group a collection of observations into clusters "such that those within each cluster are more closely related to one another than objects assigned to different clusters" as defined by Hastie, Tibshirani and Friedman [15]. It can be used to discover and identify biologically distinct subgroups from an $n \times p$ dataset without knowledge of the true group membership and plays an important role in the earlier stage of scientific studies when no definite group membership can be assigned to the observations. Clustering can also be used to establish rules for predicting memberships for additional observations, where parsimonious rules are desirable for prediction accuracy. As the definition implies, clustering algorithms are designed to maximize the between-cluster difference while minimizing the within-cluster difference. This difference, referred to as *dissimilarity*, can be measured using Euclidean distance (i.e., squared difference), Manhattan distance (i.e.,absolute dif-

ference) or other metrics depending on the problem. Typical clustering methods include k-means clustering, k-medoids clustering, hierarchical clustering and model-based clustering, with k-means clustering being one of the most popular clustering methods. It and its variants have been widely applied in various fields such as computer science, engineering, business, economics, biology, chemistry, psychology, neurology, medical imaging, pharmacology and psychiatry. It can be traced back to work done by Lloyd in 1957 [20], by Forgy in 1965 [6], by MacQueen in 1967 [21] and by Hartigan and Wong in 1979 [13], with Hartigan and Wong significantly improving the algorithm for identifying clusters. Resampling was used as an approach in "tight clustering" by Tseng and Wong in 2005 [34] to deal with noisy observations and to construct reliable tight clusters. Compared to regular clustering analysis which tends to group all n observations into clusters, tight clustering will leave out those scattered noisy samples to construct stable and reliable clusters. This was motivated by microarray studies where the goal was to find robust patterns reflecting the underlying biological process.

For a dataset, $X_{n\times p}$, with n observations and p variables per observation, earlier approaches to clustering tend to treat all of the variables equally. This may not work well when not all of the variables are equally relevant to the subgroupings, which is especially true in the case of the "high-dimension, small sample size" ($p \gg n$) problem. Clustering methods with variable selection, also called sparse clustering methods, have been recently developed to deal with this problem. Many of the sparse clustering methods make use of an L_1/Lasso penalty or related penalties to realize variable selection with Witten and Tibshirani in 2010 [35] applying both an $L1$ and an $L2$ restraint to implement variable selection. These penalties were also applied to the model-based clustering framework. Pan and Shen in 2007 [27] proposed a penalized model-based clustering method, in which BIC was used for model selection. The penalized mixture model assumes independence between variables and the same diagonal covariance matrix across clusters. Sparse clustering methods choose the variables that more accurately reflect the variability in the original dataset while diminishing or eliminating the effects of less relevant variables on the clustering. The resulting clustering rules are more parsimonious, easier to interpret, and could yield a more desirable prediction accuracy. This is true when $p < n$ but it is more evident when $p \gg n$.

In this work our goal is to deal with the problem of small sample size. It is motivated by an example arising from the identification of potentially "positive" subjects for amyloid deposition, a protein that is believed to be key in identifying subjects that will go on to develop Alzheimer's disease. The goal is to identify these subjects before they develop any of the symptoms of the disease with the hope of potential treatments to prevent the onset of symptoms. Clustering of such data is important in exploring the patterns associated with different underlying biological characteristics and in searching for a disease detection threshold. We propose a method to add resampling onto sparse clustering in order to provide a "certainty measure" for group membership.

This method combines sparse k-means and resampling, so that one can achieve variable selection and tight clusters at the same time. By resampling we will be able to more reliably select the variables and to compute a confidence level for the clustering results. This confidence level will be described in more detail in later sections. With this metric we could also construct tight clusters while leaving the outlier or noisy observations outside of these reliable clusters. We could also potentially identify the degree of the overlap between clusters. The methodology is described in Section 2. Simulation studies are shown in Section 3 to demonstrate how the proposed method works and to compare its performance to those of k-means and sparse k-means. The application of the proposed method in a neuroimaging dataset is shown in Section 4. In Section 5 we discuss the strength and limitations of the method and some problems we encountered during this work.

2.2 METHODS

2.2.1 K-means Clustering

Let X_{ij} denote a dataset of size n by p, where n is the number of observations in the dataset, p is the number of variables observed for each observation, $i = 1, ..., n$ and $j = 1, ..., p$. K-means clustering aims to group n observations into K clusters such that the within cluster dissimilarity across all clusters is minimized. The squared Euclidean distance is used as the

dissimilarity measure. The algorithm aims to minimize the within-cluster sum of squares (WCSS), so the target function is defined as

$$minimize_{C_1,C_2,\ldots C_K} \left\{ \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i \in C_k} \sum_{j=1}^{p} (x_{ij} - \mu_{kj})^2 \right\}, \tag{2.1}$$

where the clusters are denoted as $C_1, C_2, \ldots, C_K$, with n_k being the number of observations in each cluster. $\mu_k.$ is the cluster center for the k^{th} cluster, also called *centroid*, which is computed as the arithmetic average of all the observations in that cluster. μ_{kj} is the average on the j^{th} dimension.

The most widely used algorithm for k-means is the one that was developed by Hartigan and Wong in 1979. The main idea is an iterative algorithm with the following steps:

1. Set random cluster centers.
2. Assign each observation to the appropriate cluster based on the squared Euclidean distance from each cluster center.
3. Given the cluster membership, compute the new cluster centers.
4. Iterate between Step 2 and Step 3 until the cluster membership does not change any more.

It is best to use more than one set of random cluster centers in Step 1 because of the potential problem of local maxima. We assume that the number of clusters, K, can be pre-specified. If the number of clusters is in question, it can be potentially determined by *gap statistic* [33] or many other methods.

Equation (2.1) is mathematically equivalent to maximizing the between-cluster sum of squares (BCSS), that is,

$$maximize_{C_1,C_2,\ldots C_K} \left\{ \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu_{..})^2 - \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i \in C_k} (x_i - \mu_{k.})^2 \right\}, \tag{2.2}$$

where $\mu_{..}$ is the cluster center if $K = 1$, which is the average of all n observations. The average distance of all observations from this single cluster center is $\frac{1}{n} \sum_{i=1}^{n} (x_i - \mu_{..})^2$, which is also called total sum of squares (TSS).

If we calculate the values of TSS and BCSS on each dimension first and then sum across the variables, Equation (2.2) becomes:

$$maximize_{C_1,C_2,...C_K} \sum_{j=1}^{p} \left\{ \frac{1}{n} \sum_{i=1}^{n} (x_{ij} - \mu_{.j})^2 - \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i \in C_k} (x_{ij} - \mu_{kj})^2 \right\}, \tag{2.3}$$

where $\mu_{.j}$ is the value of $\mu_{..}$ on the j^{th} dimension. Equations (2.2) and (2.3) form the basis for understanding the equations for sparse k-means.

2.2.2 Sparse K-means Clustering

Sparse k-means clustering was proposed by Witten and Tibshirani in 2010 [35]. This approach minimizes the target function of the weighted BCSS with the variable weights subject to an L1/Lasso condition and an L2 condition. An EM algorithm is used for obtaining the solutions and it alternates between the weighted k-means and convex optimization with a soft-thresholding operator until convergence is reached. The results are the variable weights and cluster membership. The variable weights are assigned to the variables so that greater weight is given to the variables of greatest importance. Note that some variables have zero weights meaning that they are not used in the final clustering results. The sparse k-means clustering criteria is parsimonious for prediction and most useful when $p \gg n$ but also works well when $p < n$.

Instead of treating all variables equally as in k-means clustering, sparse k-means clustering will assign weights w_j to variables. The target function and its constraints are the following:

$$maximize_{C_1,C_2,...C_K,w} \left\{ \sum_{j=1}^{p} w_j \left(\frac{1}{n} \sum_{i=1}^{n} (x_{ij} - \mu_{.j})^2 - \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i \in C_k} (x_{ij} - \mu_{kj})^2 \right) \right\} \tag{2.4}$$

subject to

$$\forall j, w_j \geq 0, \|w\|^2 \leq 1, \|w\|_1 \leq s,$$

where

- w is a vector with length p. Each element w_j serves as a weight for each variable. All weights are nonnegative values.
- $\|w\|_1 \leq s$ is the L1/Lasso condition, meaning that the sum of the absolute values of the weights is less than s, the tuning parameter.
- $\|w\|^2 \leq 1$ is the L2 condition, meaning that the square root of the summation of the squared values of the weights is less than or equal to 1.

To solve for the weights and the cluster membership, Witten and Tibshirani [35] proposed the following iterative method:

1. Assume that all variables have the same weight $\frac{1}{\sqrt{p}}$.
2. Given the weights, find the cluster membership using the k-means algorithm. K-means is applied to the data after the variables are scaled appropriately by the squared root of the corresponding weights, which can be regarded as weighted k-means.
3. Given the cluster membership, solving for the weights can be regarded as a convex problem. To optimize the convex problem, one can apply soft-thresholding as described in Witten and Tibshirani [35].
4. Iterate between Step 2 and Step 3 until the weights converge.

The results of the sparse k-means clustering are the weights for all of the variables and the cluster membership of the individual observations. The variable weights were divided by the tuning parameter for standardization in this work. These standardized variable weights can be used for comparison between clustering results with different tuning parameters.

2.2.3 The Proposed Method

For the datasets with the property of "high-dimension, small sample size" ($p \gg n$), we propose this method that combines sparse k-means and resampling, so that one can select the variables important to clustering and construct tight clusters at the same time. At each resampling of the dataset, the variable weights and the cluster membership are obtained. These results are then summarized across all of the resampling runs.

The main idea of the proposed method is to resample the original dataset $X_{n\times p}$ for a number of times, B, and to apply sparse k-means clustering to each sample. For each resampling run, we randomly sample 70% of the observations without replacement and predict the cluster membership for the remaining 30% of the observations based on the sample results, similar as the method proposed in [34].

The algorithm is described as follows:

1. Randomly choose a sample with 70% of observations (or rows) without replacement from the original dataset $X_{n\times p}$. Let us denote this subset of the dataset X'.
2. Apply sparse k-means on X'. The number of clusters, K, should be pre-specified and remains the same for all resampling runs.
3. Apply the clustering criterion obtained from Step (2) and predict the cluster membership for the remaining 30% of the original dataset X.
4. Repeat steps (1)-(3) for a number of B times. Each variable is assigned a weight for each of the B times and the cluster membership for all n observations are obtained B times.
5. Select variables and construct tight clusters with information from Step (4). Since the tuning parameters may be different across the resampling runs, the variable weights were divided by the tuning parameter in that resampling run for standardization. The final weight for each variable can be simply computed as the average across its standardized weights from all B random samples.

   The *confidence level* is defined to be the proportion of times for an observation to appear in a cluster. For example, if the observation x_i appears in the cluster C_k for a number of b_k times out of the B resampling runs, the confidence level CL_{ik} is computed as b_k/B and $\sum_{k=1}^{K} CL_{ik} = 1$. Confidence levels for all observations are computed.

   To construct tight clusters, a threshold for the confidence level, denoted as t, should be specified. For example, $t = 95\%$ or $t = 90\%$. The observations with a confidence level $\geq t$ will be included in the tight clusters and the remaining observations will be left outside of the tight clusters. If the true underlying groups overlap with each other, the proposed method can potentially identify the degree of this overlap through the construction of the tight clusters.

2.3 SIMULATION STUDY

2.3.1 Simulation 1

We simulated data as described in Witten and Tibshirani (2010) [35]. The simulated dataset, $X_{n\times p}$, had $n = 60$ observations and p variables per observation. The sixty observations form three groups with twenty observations per group G_k, where $k = 1, 2$, and 3. These groups were generated to have different means across the three groups for the first $q = 50$ variables but not for the remaining p-q variables. That is:

- For $j = 1, 2, \cdots, q$, $X_i \in G_k \Leftrightarrow X_{ij} \sim N(\mu_k, 1)$, where $\mu_1 = -\mu$, $\mu_2 = 0$, $\mu_3 = \mu$. For the first $q = 50$ variables, the data were generated from three normal distributions with different mean values but the same variance 1.
- For $j = q + 1, \cdots, p$, $X_{ij} \sim N(0, 1)$. For the remaining p-q variables, the data were generated from the same standard normal distribution.

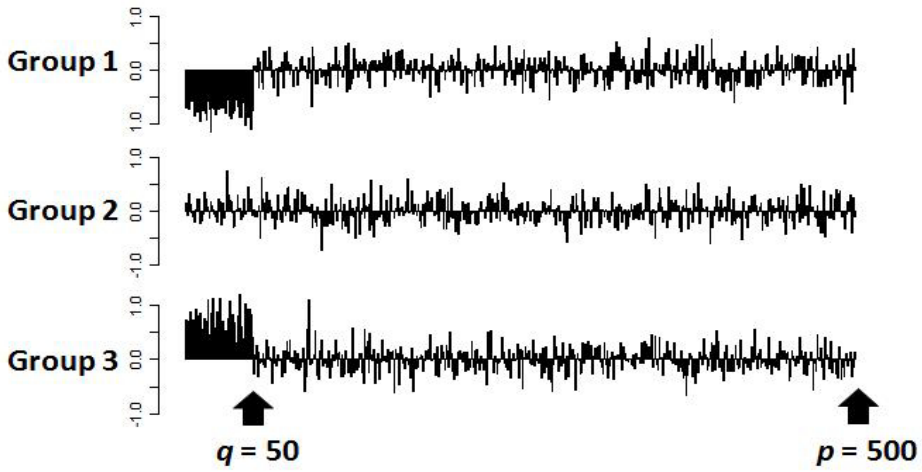


Figure 2: Group average within each variable in an example of the simulated dataset.

We chose a scenario with $p = 500$ and $\mu = 0.7$ to demonstrate how the proposed method works. We generated 250 simulated datasets and resampled each dataset $B = 20$ times.

The variable weight for the proposed method was computed as the average across the $B = 20$ resampling runs. The variable weights were then summarized over the 250 simulated

datasets. On average the first 50 variables have weights in the range of 0.012-0.016 (standard error: 0.0006-0.0018) and the remaining 450 variables have weights in 0.0005-0.0009 (standard error: 3.3×10^{-5}-0.0002). The weights for the first 50 variables were 13 to 32 times greater than those associated with the remaining uninformative 450 variables, indicating that the proposed method was able to identify the variables of greatest importance in determining cluster membership.

Table 1 shows the results for the tight clusters. The results for *one* example simulated dataset are presented in the top part of the table. The dataset was resampled 20 times. The confidence level is the observed proportion of times that an observation falls in a given cluster over the 20 resampling runs. For example, 100% indicates that 20 out of 20 times the observation was chosen to be in a specified cluster, 90% indicates that 18 out of 20 times the observation appeared in a cluster and 70% indicates that 14 out of 20 times. The confidence levels were computed for each observation for all of the clusters. With a confidence level $\geq$ 90%, there were 14 observations in cluster 1, 9 observations in cluster 2, and 8 observations in cluster 3 with one observation missclassified into cluster 1. The remaining 29 observations were left outside of these tight clusters. The three groups in the simulated dataset were overlapping because of $\mu = 0.7$. These results were consistent with this overlap. If the groups were well separated, there should be fewer observations left outside of the tight clusters. At the confidence level $\geq$ 70% and $<$ 90%, there were 7 addtional observations classified into cluster 1, 8 additional observations in cluster 2 and 11 additional observations in cluster 3 including one observation missclassified into cluster 2. If we relaxed the confidence level threhold to be $\geq$ 70% for the tight clusters, there were 21 observations in cluster 1, 17 observations in cluster 2, and 19 observations in cluster 3. Among these observations one observation in cluster 1 and another one in cluster 2 were misclassified. The remaining 3 observations were not included in any of these tight clusters.

The results for 250 simulated datasets are shown in the bottom part of Table 1. With a confidence level $\geq$ 90%, on average there were 12 observations in cluster 1, 4 observations in cluster 2, and 10 observations in cluster 3. The remaining 34 observations were left outside of these tight clusters. When the confidence level threshold was relaxed to be $\geq$ 70%, on

Table 1: Tight clusters for the scenario with $\mu = 0.7$ and $p = 500$. Top Panel: The number of observations in a cluster for one example simulated dataset. Bottom Panel: Average number of observations in tight clusters across 250 simulated datasets. Each dataset was resampled $B = 20$ times. The numbers of misclassified observations are given in the parentheses.

	One example simulation run			
confidence level	cluster 1	cluster 2	cluster 3	Remaining
$[90\% - 100\%]$	14(1)	9(0)	8(0)	29
$[70\% - 90\%)$	7(0)	8(1)	11(0)	

	250 simulation runs			
confidence level	cluster 1	cluster 2	cluster 3	Remaining
$[90\% - 100\%]$	12(0)	4(0)	10(0)	34
$[70\% - 90\%)$	6(0)	8(0)	6(0)	

average there were 6 additional observations classified into cluster 1, 8 additional observations in cluster 2, and 6 additional observations in cluster 3. The remaining 14 observations were left outside of the tight clusters. These results were also consistent with the group overlap in the simulated datasets.

Simulation 1 demonstrated how the proposed method achieves variable selection, computes the clustering confidence level, and constructs tight clusters. It was able to identify the variables which were more important to clustering and to indicate the overlap in the simulated datasets.

2.3.2 Simulation 2

This simulation is similar to Simulation 1 except that more scenarios are considered. There was one scenario with $p = 500$ and $\mu = 0.7$ in Simulation 1; In this simulation, $p = 50$, 200, 500 & 1000 and $\mu = 0.6$, 0.7, 0.8, 0.9 & 1.0, so that there were a total of 20 scenarios.

Out of the p variables, the three groups were different in the first $q = 50$ variables while not distinguishable in the remaining p-q variables. The first $q = 50$ variables were different in a way that the first group was sampled from the normal distribution $N(-\mu, 1)$, the second group from $N(0, 1)$, and the third group from $N(\mu, 1)$. The remaining p-q variables were sampled from the same standard normal distribution. The value of μ is an indicator of the overlap between the groups in the q variables. The overlap is most substantial when $\mu = 0.6$ and decreases as μ gets bigger. The groups were best separated when $\mu = 1.0$. There were 25 simulated datasets for each scenario and each dataset was resampled $B = 100$ times.

Table 2 presents the simulation results for 20 scenarios where $p = 50$, 200, 500 & 1000 and $\mu = 0.6$, 0.7, 0.8, 0.9 & 1.0. Tight clusters were constructed to include the observations with a confidence level $\geq 90\%$, which indicates that the observation appeared in the cluster at least 90 times out of $B = 100$ times. The average values were computed across the 25 simulated datasets for the number of *correctly classified* observations in each cluster and the number of remaining observations. They are listed as (cluster 1, cluster 2, cluster 3 | remaining) in Table 2. For example, for the scenario with $p = 500$ and $\mu = 0.7$, on average there were 12 observations in cluster 1, 4 observations in cluster 2 and 11 in cluster 3 and the number of remaining observations was 33.

We noticed that the method was able to identify the degree of the overlap between the groups indicating by the increasing number of remaining observations as μ gets smaller. When $\mu = 1.0$, the groups were best separated and the method was able to correctly place at least 19 observations out of 20 observations in each tight cluster with a confidence level $\geq 90\%$. As μ decreases, there were fewer observations included in the tight clusters. The smallest tight cluster was the middle group since it overlapped with the other two groups on both sides. As the number of variables, p, gets larger and μ is closer to 0, the method struggles to identify the clusters, while it performs well in settings where there is some separation.

Table 2: The average number of correctly classified observations in each cluster and the average number of remaining observations (cluster 1, cluster 2, cluster 3 | remaining) with a confidence level $\geq 90\%$. The average was computed over 25 simulated datasets in each scenario. Each dataset was resampled $B = 100$ times. The number of the truly different variables was $q = 50$.

	$p = 50$	$p = 200$	$p = 500$	$p = 1000$
$\mu = 0.6$	13,11,15|21	12,2,10|36	7,0,6|47	4,0,6|50
$\mu = 0.7$	16,15,16|13	17,13,17|13	12,4,11|33	9,1,10|40
$\mu = 0.8$	17,18,18|7	19,18,19|4	18,15,18|9	16,10,14|20
$\mu = 0.9$	19,19,18|4	20,19,19|2	20,20,20|0	19,18,19|4
$\mu = 1.0$	19,19,19|3	20,19,20|1	20,19,20|1	20,20,20|0

2.3.3 Methods Comparison

The goal of this simulation study was to compare three clustering methods: k-means, sparse k-means and the proposed method "sparse k-means and resampling". We use all of the datasets generated in Simulation 2 for all 20 scenarios. K-means, sparse k-means and the proposed method were applied to these datasets with $K = 3$. The clustering results are summarized in Tables 3 and 4. For sparse k-means and the proposed method, the tuning parameter pool was the same in each scenario, which is a vector of 15 values within the range $(3.5,\ 0.7 \times \sqrt{p})$.

The performance of clustering was measured by the *classification error rate* or CER [35]. In this simulation study, it is a measure of disagreement between the true group membership and the cluster membership. The CER value can range from 0 to 1. A classification error rate with value 0 indicates a perfect agreement, or no disagreement, between the group membership and the cluster membership. The bigger the CER value, the less accurate the clustering recovered the true group membership.

Table 3 lists the average classification error rate (CER) values along with their standard errors across the 25 simulated datasets in each scenario. The results for each of the three methods were given. For the proposed method, this CER value was calculated for the subset of those observations in the tight clusters with confidence level $\geq 90\%$. We used the classification error rate for the comparison of these methods so that others can relate our results with those in Witten and Tibshirani (2010) [35], but we were aware of the limitations of this measure. In the simulation study the true group membership were assigned when the data were generated. If there was overlap between groups, the observations mixed in the overlap were given different group membership; When clustering was applied, it was blind to the true group memberhsip and could hardly disentangle this mixed overlap back into the correct clusters, so it was almost impossible to achieve clustering results which perfectly agrees to the true group memberhsip. With the tight clusters constructed by the proposed method, we were able to achieve smaller classification error rates because of the fact that it was calculated for the observations in the tight clusters only, which were usually not part of the overlap.

When $p = q = 50$ in Table 3, since all of the 50 variables were different and they were different in the same way between the groups, k-means achieved smaller CER values than sparse k-means. Sparse k-means might have tried to assign different weights to the variables, which could be the reason for the smaller CER values. It is worthwhile to point out, although it is not the case here, that if the group difference varies across the variables, sparse k-means may be a better method than k-means. When $q = 50$ and $p = 200, 500$ and 1000, there were only a subset of q variables that were different across groups. The CER values for sparse k-means were smaller than those from k-means. This was because under these scenarios while k-means treated all of the variables the same way, sparse k-means was able to discover the variables important to clustering and to leave out the irrelevant variables. For all twenty scenarios the proposed method achieved even smaller CER values than the other two methods, this was because the CER value was computed for the subset of those observations in the tight clusters. These very small CER values (range: 0.000-0.032) indicates that the observations in the tight clusters were recovered from the corresponding true groups with a very high accuracy.

Table 3: The average classification error rate (standard error) over 25 simulated datasets for each scenario. For our proposed method "sparse k-means and resampling", each dataset was resampled $B = 100$ times and the classification error rates were calculated for those observations with a confidence level $\geq 90\%$.

k-means

	$p = 50$	$p = 200$	$p = 500$	$p = 1000$
$\mu = 0.6$	0.067(0.008)	0.165(0.013)	0.234(0.011)	0.283(0.006)
$\mu = 0.7$	0.027(0.006)	0.080(0.007)	0.190(0.012)	0.225(0.008)
$\mu = 0.8$	0.005(0.002)	0.031(0.005)	0.099(0.011)	0.189(0.012)
$\mu = 0.9$	0.002(0.001)	0.013(0.005)	0.038(0.006)	0.101(0.010)
$\mu = 1.0$	0.001(0.001)	0.002(0.001)	0.009(0.003)	0.044(0.006)

sparse k-means

	$p = 50$	$p = 200$	$p = 500$	$p = 1000$
$\mu = 0.6$	0.133(0.011)	0.137(0.014)	0.197(0.013)	0.252(0.011)
$\mu = 0.7$	0.067(0.008)	0.056(0.006)	0.087(0.016)	0.136(0.017)
$\mu = 0.8$	0.040(0.005)	0.017(0,005)	0.025(0.006)	0.045(0.011)
$\mu = 0.9$	0.015(0.004)	0.009(0.004)	0.006(0.002)	0.008(0.002)
$\mu = 1.0$	0.008(0.003)	0.001(0.001)	0.001(0.001)	0.002(0.001)

sparse k-means and resampling

	$p = 50$	$p = 200$	$p = 500$	$p = 1000$
$\mu = 0.6$	0.032(0.007)	0.016(0.006)	0.008(0.005)	0.004(0.004)
$\mu = 0.7$	0.009(0.003)	0.006(0.002)	0.003(0.002)	0.015(0.007)
$\mu = 0.8$	0.003(0.002)	0.000(0.000)	0.001(0.001)	0.002(0.001)
$\mu = 0.9$	0.001(0.001)	0.003(0.002)	0.002(0.001)	0.002(0.001)
$\mu = 1.0$	0.000(0.000)	0.000(0.000)	0.000(0.000)	0.000(0.000)

Table 4: The average number (standard error) of variables with weights $> 1/p$. The average was computed over 25 simulated datasets for each scenario. For our proposed method "sparse k-means and resampling", each dataset was resampled $B = 100$ times. The number of variables truly different between groups was $q = 50$.

sparse k-means

	$p = 50$	$p = 200$	$p = 500$	$p = 1000$
$\mu = 0.6$	20.12(0.31)	57.88(0.58)	79.20(5.38)	54.80(5.18)
$\mu = 0.7$	20.44(0.41)	53.24(0.40)	80.28(3.18)	66.80(3.63)
$\mu = 0.8$	20.96(0.24)	51.84(0.44)	67.68(1.49)	58.52(3.06)
$\mu = 0.9$	21.32(0.34)	50.76(0.28)	64.48(1.20)	73.92(4.51)
$\mu = 1.0$	20.60(0.28)	50.00(0.08)	60.84(1.33)	83.44(3.18)

sparse k-means and resampling

	$p = 50$	$p = 200$	$p = 500$	$p = 1000$
$\mu = 0.6$	20.76(0.40)	50.96(0.46)	63.92(2.13)	102.00(4.25)
$\mu = 0.7$	21.20(0.42)	52.44(0.42)	59.00(1.12)	62.72(1.16)
$\mu = 0.8$	20.76(0.30)	51.48(0.37)	55.52(0.73)	56.80(0.65)
$\mu = 0.9$	21.04(0.35)	50.64(0.25)	55.80(0.51)	55.60(0.53)
$\mu = 1.0$	20.92(0.30)	50.04(0.09)	54.84(0.42)	51.76(0.35)

Table 4 lists the average number (and its standard error) of variables with the final standardized weights $> 1/p$. It was summarized from results with sparse k-means and the proposed method "sparse k-means and resampling". Recall from the methods section, for sparse k-means the final variable weight was the standardized weight, which was the original variable weight divided by tuning parameter; for the proposed method the final variable weight was the average standardized weight over $B = 100$ resampling runs. Because of the standardization, the final weights were comparable between clustering results with different tuning parameters. If all of the variables were equally important to the clustering, each variable would have been assigned the same weight $1/p$. We counted the number of the variables with final weights $> 1/p$, which were also the variables that affect clustering more than they would have if all of the p variables were treated equally.

Recall that the number of variables that were truly different between groups was $q = 50$ while the number of variables $p = 50$, 200, 500, and 1000. When $p = 50$, both of the sparse k-means and the proposed method assigned weights $> 1/p$ to about 20 variables and weights $\leq 1/p$ to the remaining variables. These methods were not a good choice here since all of the $q = 50$ variables were generated to be similarly different between groups. When $p = 200$, 500, and 100, the proposed method were able to more closely identify important variables in that the number of variables with weights $> 1/p$ was closer to the number $q = 50$ than that of sparse k-means in most of scenarios. An acceptable exception was the scenario with $\mu = 0.6$ and $p = 1000$ because the aforementioned problem with this scenario. Overall the proposed method was able to identify the variables with greatest importance to clustering and its performance was better than that of sparse k-means in scenarios where p gets larger.

2.4 APPLICATION IN NEUROIMAGING DATA

We apply the proposed method to a brain imaging dataset. At first we will describe the background for this application. According to the Alzheimer's Association 2011 Alzheimer's Disease Facts and Figures, the neurological degenerative disease, Alzheimer's Disease (AD), counts for about 60-80% cases of dementia. It is estimated that "5.4 million Americans

are living with" the disease. It is "the $6th$ leading cause of death" in the country and "the only cause of death among the top 10 that cannot be prevented, cured or even slowed" (http://www.alz.org/). *Amyloid-*$\beta(A\beta)$ plaque deposition is a hallmark for AD neuropathology. Studies have shown that the amyloid deposition could have started more than a decade before the onset of any memory symptoms. It is of great interest to detect those people who are at risk of developing AD. Early detection of these individuals may allow a slowing or stoppage of the progress of the disease long before its onset.

The definitive diagnosis of AD can only be given by pathological tests after the patient has deceased; the ante-mortem diagnosis is "Probable AD". The development of $[^{11}C]$ Pittsburgh compound-B (PiB) PET imaging agent allowed imaging of *Amyloid-*$\beta(A\beta)$ deposition in living humans in 2003 [19, 22]. PiB has been widely studied by many institutions and large-scale studies are under way in the United States, Japan and Australia. It can potentially serve as an *in-vivo* test to monitor the progress of amyloid deposition and a surrogate measure for treatment effectiveness. Developing thresholds for detecting those cognitively normal people who are at risk of developing AD based on the data from the PiB PET scans has been an ongoing research focus among the Pittsburgh Amyloid Imaging group, whom we collaborate closely with.

Establishing the AD early detection diagnostic criteria is facing crucial challenges due to the lack of information for a reference standard. To avoid confusion, it is worthwhile pointing out again that this threshold is going to be established and used for *cognitively normal* subjects only; this is different than the threshold used to differentiate controls and AD patients. Then the PiB early detection threshold should also be validated by the post-mortem pathological tests. But technical challenges remain in the area of combining and correlating the ante- and post-mortem data. Even if these technical challenges were solved, there are only a very small subset of subjects with both ante- and post-mortem data. Longitudinal follow-up may also serve as a validation of the threshold. But not enough longitudinal scans have been collected to date to make this approach feasible. Multiple studies are ongoing to collect all these data but due to the fact that amyloid accumulation is very slow, acquiring sufficient data will still take a while. With these challenges in mind, efforts have been made to establish the AD early detection diagnostic criteria based on baseline brain imaging data.

The goal of this example was to identify cognitively normal elderly people who were amyloid positive based on their brain imaging data. The dataset consisted of 64 normal control elderly subjects (43 female, 21 male) with an average age of 74 (standard deviation = 5.4; range: 65-89). A brain image was acquired for each subject with the value of each voxel reflecting the amount of amyloid deposition in the brain. There are $343,147$ voxels in each image after the non-brain voxels were masked out. The input for the clustering analysis was a dataset with $n = 64$ subjects and $p = 343,147$ voxels per image for the analysis. We intended to classify the dataset into two clusters, so K is equal to 2. We applied the proposed method and the dataset were resampled 50 times.

A brain map of voxel (or variable) weights, which indicates their importance in distinguishing clusters, was computed for each of these samples during the resampling. These weights were standardized for the different tuning parameters, which ranged from 305.56 to 383.89. A total of 50 brain maps were generated with one for each resampling run. As described in the methods section, the final weight for each voxel was computed as the average of the standardized weights across the 50 resampling runs. The image with the final voxel weights is shown as the right half in Figure **??**, in which a brighter color means a bigger weight while a darker color means a smaller weight. On the left a high resolution Magnetic Resonance (MR) Image showed the corresponding brain anatomy. The slices shown here were representative of typical regions with amyloid deposition.

The range of the weights is from 0 to 9.86 fold of $1/p$, where $p = 343,147$ and $1/p = 2.88 \times 10^{-6}$. The voxel weights were bigger in the prefrontal, the precuneus, and the parietal cortex regions indicating that these regions were important for subgroupings while the voxel weights in the sensorimotor and cerebellar cortex were much smaller indicating they were not that important for the subgroupings. These findings were consistent with the distribution of amyloid deposition in the human brain across a spectrum of subjects and biologically meaningful.

Among all 64 subjects, there were 7 subjects identified as amyloid positive and 37 subjects identified as amyloid negative at the confidence level 100%. Another subject was identified as positive at a confidence level 84%. The remaining 19 subjects were negative at the confidence levels 98% (for four subjects), 96%(for one subject), 94% (for two subjects), 90% (for three

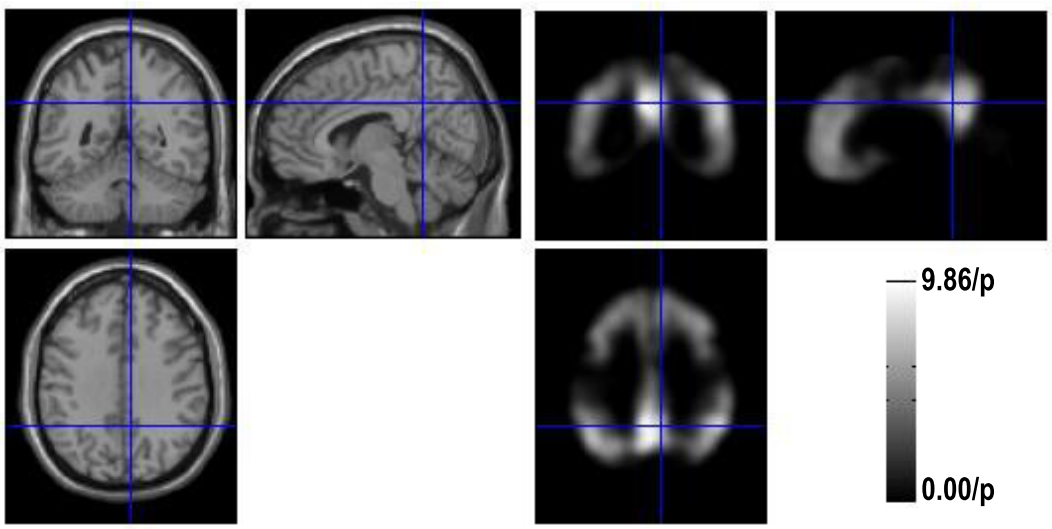


Figure 3: Voxel/Variable weights. (Left) Magnetic Resonance Image which shows the brain anatomy; (Right) The image with the final weights for each voxel (or each variable). The range of weights is 0 to 9.86 fold of $1/p$, where $p = 343,147$ and $1/p = 2.88 \times 10^{-6}$. The image dataset was resampled 50 times.

subjects), 88% (for four subjects), 84% (for one subject), 78% (for one subject), and 76% (for three subjects). These results indicates that there were 7 subjects who were always amyloid positive and 1 subject somewhat positive, and there were 37 subjects who were always amyloid negative, 5 who were very negative with confidence levels $> 95\%$, and 14 who were somewhat negative with a confidence level in the range of $76\% - 94\%$. In spite of the fact that the amyloid groups were not clearly separated because of these somewhat positive/negative subjects, all of the subjects were classified into a group with strong confidence with the confidence levels being $\geq 76\%$ for the negative subjects and $\geq 84\%$ for the positive ones. This may serve as evidence against the idea of a third cluster for these intermediate subjects. To further understand a potential classification of "amyloid positive", we could examine the characteristics of the dataset with the information provided by the clustering results. For example, we could assess the difference between the somewhat positive subject against the group with seven always positive subjects or against the group with the fourteen somewhat negative subjects. We could also choose to compute cluster centers from the tight clusters with a confidence level threshold $\geq 95\%$ and use these reliable cluster centers to predict membership for any additional PiB scans.

2.5 CONCLUSIONS AND DISCUSSION

The proposed method, "sparse k-means and resampling", can achieve both variable selection and tight clusters for datasets with the property of "high-dimension, small sample size" ($p \gg n$). The final standardized variable weights can serve as a more reliable metric to exclude the irrelevant variables and yield parsimonious clustering results. The confidence levels for the clustering results provided more information than the usual approach of dichotomizing results from clustering methods without resampling. These confidence levels can be used to construct tight clusters with different degrees of tightness and to potentially identify the degree of the overlap between the clusters. To date, this work is the first method to combine resampling with sparse clustering. This method is especially useful for neuroimaging datasets, where the small sample size is usually a concern.

The proposed method provides additional information on variable weights and clustering results through resampling. Bootstrapping was not used because bootstrapping resamples the data with replacement, which makes it very complicated to summarize the results across resampling runs. We illustrated the use of metrics such as the final standardized variable weight and the confidence level for the cluster results. Depending on the goal of the scientific question, other metrics may be more appropriate for summarizing the resampling results. For example, in the case of constructing rules for diagnostic tests, it may be of interest to find more reliable cluster centers and to use them for cluster membership prediction. In this case, the focus will be summarizing the cluster centers from the resampling results. Like sparse k-means, this method is most useful when $p \gg n$, but also works in the case of $p < n$.

One major limitation of this method is that resampling tends to be time-consuming. It is an inevitable trade off that we have to make to gain more confidence in the clustering results. The method does not perform as well when the underlying groups significantly overlap with each other and the number of variables, p, is much larger than the number of truly different variables, q. Prescreening of variables may be useful to decrease the number p before clustering is applied, and is a topic for future research. We have showed that the proposed method works well when the number of clusters is 2 or 3. We anticipate that its performance may be compromised when the number of clusters gets larger. For example, it may become very challenging to properly align the cluster centers from the different resampling runs.

We have also encountered the problems related to the tuning parameter choice based on gap statistic. The Witten and Tibshirani (2010) [35] paper pointed out that the performance of the gap statistic for selecting the tuning parameter was not always great. In our experience the gap statistic has worked well in choosing the tuning parameters in neuroimaging applications. But when the default tuning parameter pool $\left(1.2, 0.9\sqrt{p}\right)$ was used in Simulations 1 and 2, the tuning parameters that were either too small, or too large, yielded clustering results with unusually higher classification error rates. This was why we chose to use an adjusted tuning parameter pool in this paper. We have also been searching for methods that could potentially address this issue.

In summary, the proposed method of combining sparse clustering with resampling provides an improvement over currently existing techniques. The use of a confidence metric, based on the observed proportion of times that an observation falls in a given cluster is extremely useful for assessing the potential classification of normal control subjects into "amyloid positive" and "amyloid negative" groups. These groupings are key for targeting future prevention strategies in Alzheimer's disease.

3.0 TUNING PARAMETER CHOICE FOR SPARSE K-MEANS CLUSTERING

3.1 INTRODUCTION

Clustering is important in exploring and discovering distinct biological groups in the earlier stage of scientific studies when no definite group membership could be assigned to the observations. It can be used to establish rules for predicting memberships for additional observations. Clustering methods with variable selection, also called sparse clustering, choose the variables that more accurately reflect the variability in the original dataset while diminishing or eliminating the effects of less relevant variables on the clustering. The resulting clustering rules are more parsimonious, easier to interpret, and could yield a more desirable prediction accuracy. This is true when $p < n$ but it is more evident in the case of "high-dimension, low sample size" ($p \gg n$).

Variable selection realized by penalties was first developed in linear regression. A very influential approach was *Lasso* [29], which stands for "Least absolute shrinkage and selection operator". It was proposed by Tibshirani in 1996 and has been widely applied since then. For linear regression, an L1/Lasso penalty imposes an upper bound s to the summation of absolute values of the regression coefficients β_i's, i.e. $\sum_{i=1}^{p} |\beta_i| \leq s$, where p is the number of independent variables in the model. The algorithm minimizes the target function of the sum of squares with restrictions of this penalty. The resulting coefficients are then subject to shrinkage in absolute value (absolute shrinkage) or have a value of 0 (selection). When the coefficient is 0, the variable is not selected in the model. Solving for the problem can be regarded as convex optimization and solutions are obtained by applying the soft-thresholding operator because it satisfies the Karush-Kuhn-Tucker conditions [2, 29]. Least

Angle Regression (LARS) was developed by Efron, Hastie, Johnstone and Tibshirani in 2002 to solve the Lasso in a much faster manner [5]. A further significant improvement in solving Lasso was coordinate descent algorithm [11, 9, 10, 37, 12]. These algorithms have evolved to be so efficient that the computing time is no longer a concern.

Lasso and related penalties have been applied to a wide range of traditional statistical methods to realize variable selection. The resulting methods are called *regularization* methods. They have been applied to linear regression [29], generalized linear models [29], survival analysis [30], clustering [35, 27] and support-vector machine [14]. They have also been applied to dimension reduction methods such as principal component analysis [17, 36], canonical correlation analysis [36] and partial least squares [4]. Variants of penalties include the Grouped Lasso [38, 26], Elastic Net [41], Adaptive Lasso [40], Graphical Lasso [39, 9], Dantzig Selector [3] and others. A discussion of these penalties was given in [10, 31]. In this work we will focus on the application of these penalties in clustering and the tuning parameter choice.

Most of the clustering methods with variable selection are implemented with an $L1$ or *Lasso* penalty. A general framework for clustering was proposed by Witten and Tibshirani in 2010 [35]. It applied an $L1$ and an $L2$ restraint on k-means to realize variable selection, i.e. sparse clustering. The algorithm was implemented by an EM algorithm alternating between weighted k-means and convex optimization. The authors showed that it also works for hierarchical clustering and k-mediods clustering. These penalties were also applied in the model-based clustering framework. Pan and Shen in 2007 [27] proposed a penalized model-based clustering method, in which a *Lasso* penalty and a modified BIC were used to realize model selection. The penalized mixture model assumes independence between variables and the same diagonal covariance matrix across clusters.

The sparse clustering method results depend on the choice of the tuning parameter [35] or the penalization parameter [27] (For simplicity reasons, we will only use the term "tuning parameter" from here on). For example in the Witten and Tibshirani paper [35] a larger tuning parameter usually allows for more of the variables to be selected for clustering and vice versa. The choice of the tuning parameter has been studied in regression and classification with cross-validation being the main approach. This method can not be easily

applied to clustering due to the lack of the true membership. Pan and Shen in 2007 [27] used a modified BIC to determine both the number of clusters and the tuning parameter. Witten and Tibshirani in 2010 [35] proposed using the gap statistic. Yet both of these methods rely on the correct specification of the tuning parameter pool.

We have observed that given a suboptimal pool the gap statistic method could choose a tuning parameter which yields poor performance when clustering, i.e. a high Classification Error Rate (CER). When this happens it usually chooses a tuning parameter that is too small or too large, hence too few, or too many, variables are chosen for clustering. This is more likely to occur when the overlap between the clusters is more substantial. In this work, we would like to draw attention to this phenomenon and also to propose using BIC for the choice of tuning parameter for the sparse k-means method proposed by Witten and Tibshirani in 2010.

3.2 GAP STATISTIC FOR TUNING PARAMETER CHOICE

The gap statistic was originally proposed by Tibshirani in 2001 [33] for the purpose of choosing the number of clusters when clustering is applied to a dataset and the number of clusters is unknown. The Witten and Tibshirani[2010] paper adopted it for the purpose of choosing the tuning parameter [35]. The definitions and notations in these two papers are different although the essence of it is the same. The gap statistic is defined to be the difference, i.e. *gap*, between the observed value and the expected value of the target function on the logarithm scale. In the Witten and Tibshirani[2010] paper, the target function in sparse k-means is to maximize the weighted Between-Cluster Sum of Squares (BCSS), which is defined as

$$O(s) = \sum_{j=1}^{p} w_j \left(\frac{1}{n} \sum_{i=1}^{n} \sum_{i'=1}^{n} d_{i,i',j} - \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i,i' \in C_k} d_{i,i',j} \right). \tag{3.1}$$

The corresponding gap statistic is defined as

$$gap(s) = \log(O(s)) - E\left[\log(O_b(s))\right], \tag{3.2}$$

where $E\left[\log(O_b(s)\right]$ is the expected value of the target function on the logarithm scale. The expected target function value is obtained by permutation and will be discussed in detail next.

The permuted dataset is generated in a way such that the values in each column are randomly chosen from the corresponding column in the original dataset. There are B permuted datasets. These permuted datasets are supposed to reflect the reference, or the null, distribution. For a given tuning parameter, sparse clustering will be applied to each of these datasets and a value of the target function can be calculated. The average of them $\frac{1}{B}\sum_{b=1}^{B}\log(O_b(s))$ is the expected value of the target function. Simply put, it is the average of the weighted BCSS for the B permuted dataset. Thus the gap statistic is

$$gap(s) = \log(O(s)) - \frac{1}{B}\sum_{b=1}^{B}\log(O_b(s)). \tag{3.3}$$

We will compute the gap statistic for every choice in the tuning parameter pool. Intuitively, the tuning parameter s^* for which the clustering achieves the largest gap statistics is chosen for sparse clustering. The default pool for tuning parameter choice is $\left(1.2, 0.9\sqrt{p}\right)$.

3.3 TUNING PARAMETER CHOICE IN PENALIZED MODEL-BASED CLUSTERING

3.3.1 Model-based Clustering

For a dataset, $X_{n\times p}$, with n observations and p variables observed for each observation, suppose that the number of clusters is K. Model-based clustering assumes that each observation of the dataset is drawn from a finite mixture distribution. Assume that each underlying cluster C_k follows a multivariate distribution $f_k(x;\theta_k)$, where θ_k is a vector of the parameters specifying the distribution. The K clusters form a mixture of these multivariate distributions, which is a finite mixture distribution and can be denoted as $f(x;\theta) = \sum_{k=1}^{K}\pi_k f_k(x;\theta_k)$. Here π_k is the probability of one observation belongs to the kth cluster and $\sum_{k=1}^{K}\pi_k = 1$. The clustering algorithm is to find the cluster memberships and the distribution parameters,

θ_k, which maximize the likelihood of observing the dataset. The log-likelihood can be written as

$$\log L(\theta) = \sum_{i=1}^{n} \log[\sum_{k=1}^{K} \pi_k f_k(x_i; \theta_k)], \tag{3.4}$$

where x_i, $i = 1, 2, \ldots, n$, is the ith observation in the dataset, $X_{n \times p}$.

EM algorithms are used to obtain the solutions as described in McLachlan and Peel[2002] [23, 24] and Fraley and Raftery[2002] [8].

3.3.2 Penalized Model-based Clustering

Under the model-based clustering framework and assuming a multivariate normal distribution the Pan and Shen [2007] paper proposed a penalized model-based clustering approach [27]. It assumes that the dataset is sampled from a finite mixture of multivariate normal distributions. The method further assumes that the covariance for each component is a diagonal matrix and they hold the same across all components. Recall that the p-dimensional multivariate normal distribution density with mean μ and covariance Σ is

$$f(x; \mu, \Sigma) = \frac{1}{(2\pi)^{p/2} |\Sigma|^{\frac{1}{2}}} exp \left[-\frac{1}{2}(x - \mu)' \Sigma^{-1} (x - \mu) \right]. \tag{3.5}$$

Hence, each component of the mixture distribution can be denoted as

$$f_k(x; \mu_k, V) = \frac{1}{(2\pi)^{p/2} |V|^{\frac{1}{2}}} exp \left[-\frac{1}{2}(x - \mu_k)' V^{-1} (x - \mu_k) \right], \tag{3.6}$$

where μ_k is the mean for the kth component; the covariance matrix is the same for each component, it is $V = diag(\sigma_1^2, \sigma_2^2, \ldots, \sigma_p^2)$ and $|V|^{\frac{1}{2}} = \prod_{j=1}^{p} \sigma_j$.

The penalized log-likelihood is

$$\log L(\theta) = \sum_{i=1}^{n} \log[\sum_{k=1}^{K} \pi_k f_k(x_i; \theta_k)] - h_\lambda(\theta), \tag{3.7}$$

where $h_\lambda(\theta) = \lambda \sum_{k=1}^{K} \sum_{j=1}^{p} |\mu_{kj}|$. This L_1-penalty term is imposed so that the mean values μ_{kj} for the finite mixture distribution are subject to shrinkage. Similarly an EM algorithm is used to solve for the clustering memberships and the distribution parameters. Details are described in [27]. When all the μ_{kj} values for the same j are estimated to be zeroes, the jth parameter has no effect on clustering, i.e. it is not selected in the final clustering model.

3.3.3 BIC for Tuning Parameter Choice

The Fraley and Raftery [1998] paper [7] proposed to use Bayesian Information Criterion (BIC) [28] for model selection in the framework of model-based clustering, where

$$BIC = -2 \log L(\hat{\Theta}) + d \log(n), \tag{3.8}$$

in which $\hat{\Theta}$ is the Maximum Likelihood Estimator (MLE) and d is the number of parameters to estimate.

For penalized model-based clustering since the models being compared may have different numbers of parameters due to the degree of regularization, i.e. the value of the tuning/penalization parameter, a modified BIC is used for model selection. The modified BIC is defined as

$$BIC_e = -2 \log L(\tilde{\Theta}) + d_e \log(n), \tag{3.9}$$

where $\tilde{\Theta}$ is the Maximum Penalized Likelihood Estimator (MPLE) and d_e is the number of parameters to estimate in the regularized model. Since the parameters to estimate without regularization are π_k's, σ_j's, and μ_{kj}'s and $\sum_{k=1}^{K} \pi_k = 1$, the number of parameters is $K + p + Kp - 1$. The number of free parameters is $d_e = K + p + Kp - 1 - q$ where q is the number of zero μ_{kj}'s.

The modified BIC BIC_e was used to determine both the number of clusters and the tuning parameter. A tuning parameter pool will be specified as a vector with N_λ possible tuning parameter values and a small possible number of K's, say N_K, will also be pre-specified. Then penalized model-based clustering will be fitted to the dataset with every of the $N_\lambda \times N_K$ scenarios. The model with the best BIC will be chosen from all these $N_\lambda \times N_K$ models.

3.4 PROBLEMATIC RESULTS USING GAP STATISTIC

The Witten and Tibshirani[2010] paper pointed out that "the performance of the gap statistic" "for selecting the tuning parameter is mixed" [35]. In our experience the gap statistic has worked well in choosing the tuning parameters in neuroimaging applications either when $p < n$ for a dataset with regional outcome measures or when $p \gg n$ with voxel-wise outcome measures. But when the default tuning parameter pool $\left(1.2, 0.9\sqrt{p}\right)$ was used in the first simulation in the Witten and Tibshirani[2010] paper, the tuning parameters were chosen to be too small or too large in certain scenarios that yielded clustering results with higher CER values. We will dissect the problem in more detail next.

This simulation set up is similar to that outlined in section 2.3.2. The simulation was originally chosen [35] to compare sparse k-means and k-means. There are three groups in the original dataset with 20 observations in each group. The groups were different in the first $q = 50$ variables while not distinguishable in the remaining $p - q$ variables, where $p =$ 50, 200, 500 and 1000. The first $q = 50$ variables were different in a way that the first group was sampled from the normal distribution $N(-\mu, 1)$, the second group from $N(0, 1)$, and the third group from $N(\mu, 1)$, where $\mu = 0.6$, 0.7, 0.8, 0.9 and 1.0. The value μ is an indicator of the overlap between the groups in the q variables. The overlap is most substantial when $\mu = 0.6$ and decreases as μ gets bigger. The groups were best separated when $\mu = 1.0$.

Table 5: Tuning parameter choice for $\mu = 0.6$ and $p = 500$

Simulation run	1	2	3	4	5	6	7	8	9	10
Tuning parameter	14.71	7.86	5.75	14.71	10.75	10.75	14.71	10.75	7.86	1.2
# of nonzero weights	500	150	89	500	321	306	500	339	155	4
CER	0.23	0.25	0.27	0.29	0.18	0.29	0.27	0.16	0.30	0.38
Simulation run	11	12	13	14	15	16	17	18	19	20
Tuning parameter	10.75	14.71	1.2	10.75	1.2	7.86	14.71	14.71	1.2	7.86
# of nonzero weights	369	500	7	336	5	145	500	500	5	155
CER	0.28	0.28	0.38	0.25	0.37	0.23	0.30	0.26	0.38	0.12

The sparse k-means method proposed by Witten and Tibshirani requires that the tuning parameter is in the range $\left[1, \sqrt{p}\right]$ [35]. The default tuning parameter pool is specified as $\left(1.2, 0.9\sqrt{p}\right)$ for the sparse k-means R routine implemented by the authors with the default number of tuning parameters in the pool being 10. The tuning parameters were chosen

Table 6: Tuning parameter choice for $\mu = 0.6$ and $p = 1000$

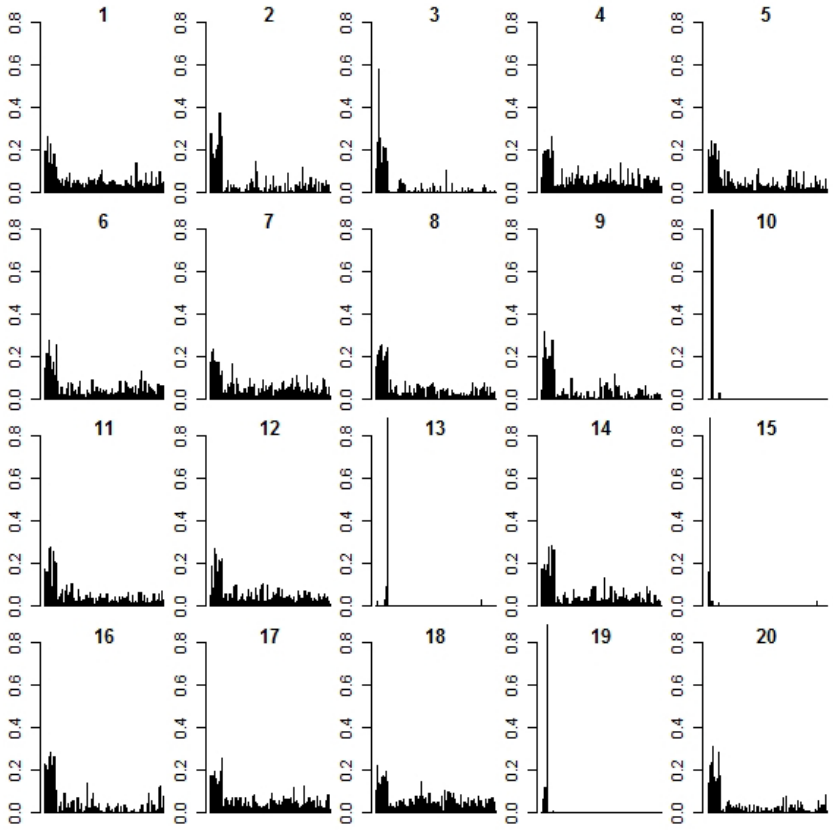
Simulation run	1	2	3	4	5	6	7	8	9	10
Tuning parameter	2.43	9.91	4.90	9.91	14.08	1.2	6.97	1.2	3.45	1.20
# of nonzero weights	13	288	59	250	537	2	118	7	27	5
CER	0.32	0.21	021	0.28	0.26	0.39	0.14	0.38	0.25	0.41
Simulation run	11	12	13	14	15	16	17	18	19	20
Tuning parameter	9.91	1.20	1.20	9.91	6.97	6.97	1.20	6.97	6.97	1.20
# of nonzero weights	242	6	4	256	142	109	7	133	127	6
CER	0.27	0.44	0.39	0.28	0.27	0.10	0.36	0.27	0.26	0.42

in a way such that they were equally spaced in the log-scale and the numbers were then transformed back by applying an exponential function, so the tuning parameters in the default pool are 1.20, 1.64, 2.25, 3.07, 4.20, 5.75, 7.86, 10.75, 14.71 and 20.12 when $p = 500$. When $p = 1000$ the tuning parameters in the default pool are 1.20, 1.71, 2.43, 3.45, 4.90, 6.97, 9.91, 14.08, 20.02 and 28.46.

Using this default pool, we found that when the overlap between groups was more substantial such as scenarios "$\mu = 0.6$ & $p = 500$" and "$\mu = 0.6$ & $p = 1000$" the gap statistic could choose tuning parameters that were either too small such as 1.2 or 2.43 , or too large such as 14.71 or 20.12. This also happened in the scenarios $\mu = 0.7$ and $p = 500$ but it happened much less often at a probability of ≤ 0.05.

Table 5 shows the tuning parameters, the number of variables with nonzero weights and CER values for 20 simulation runs. When the tuning parameter was chosen to be a large value, 14.71, in simulation runs No. 1, 4, 7, 12, 17 and 18, the number of variables with nonzero weights was 500 for all these simulation runs and the CER values had an average of 0.27 and ranged $0.23 - 0.30$. When the tuning parameter was chosen to be small, 1.2, in simulation runs No. 10, 13, 15 and 19, the number of variables with nonzero weights ranged $4 - 7$ and CER values were obviously higher with an average of 0.378 with the individual values ranging $0.37 - 0.38$. For the tuning parameters in the middle range in the remaining ten simulation runs, the number of variable with nonzero weights ranged $89 - 369$ and the CER values had an average of 0.23 and ranged $0.12 - 0.30$. Further examination of the variable weights in Figure **??** showed that when the tuning parameter 1.2 was chosen at

Figure 4: Variable weights in a problematic scenario with the gap statistics. Weights for 500 variables when where $\mu = 0.6$ are plotted for 20 simulation runs.



least one of the nonzero variables was assigned a very large weight of ≥ 0.8. In contrast the remaining variables in these simulation runs and all variables in other simulation runs were assigned with weights much smaller and they were around or less than 0.2. These extra large nonzero weights skewed the clustering rules by casting too much weight on these variables and caused high CER values. When the tuning parameter 14.71 was chosen, all 500 variables were chosen for clustering and it may be that the irrelevant variables caused the CER values to be higher than those obtained with moderate tuning parameters.

Table 6 shows the tuning parameters and the number of variables with nonzero weights for 20 simulation runs. When the tuning parameter was chosen be to large, 14.08, in the $5th$ simulation run, the number of variables with nonzero weights was 537 and the CER value

was 0.26. When the tuning parameter was chosen to be small (1.2 or 2.43) in simulation runs No. 1, 6, 8, 10, 12, 13, 17, and 20, the number of variables with nonzero weights ranged $2-13$ and CER values were obviously higher with an average of 0.389 with the individual values ranging $0.32-0.44$. For the tuning parameters in the middle range in the remaining eleven simulation runs, the number of variable with nonzero weights ranged $27-288$ and the average CER value was 0.23 and ranged $0.10-0.28$. The variable weights showed similar pattern for tuning parameters 1.2 and 2.43 as exhibited in Figure **??**.

In summary these results show that for these scenarios $\mu = 0.6$ and $p = 500$ or 1000, the small tuning parameters (1.2 or 2.43) may need to be avoided given that the variable selection was inaccurate which resulting the clustering results with obviously larger classification errors. The large tuning parameter allowed for too many variables chosen in clustering which resulting bigger CER values because of the subgrouping-irrelevant variables in the model. This shows that we should consider restricting the tuning parameter pool from both ends although justifications should be made while doing so. It also seemed that the lower bound of tuning parameter pool should be chosen to be proportional to the number of variables p rather than a fixed number 1 or 1.2. The reason is that when the number of variables p is very large, even though the variable weights could be very small, the summation of these values could easily bigger than 1 or 1.2.

The first simulation in the Witten and Tibshirani[2010][35] paper did not utilize the default tuning parameter pool $\left[1.2, 0.9\sqrt{p}\right]$. Instead an adjusted tuning parameter pool $\left[2, 0.7\sqrt{p}\right]$ with 15 values was used. The pool included the following numbers: 2.00, 2.32, 2.68, 3.11, 3.60, 4.17, 4.83, 5.60, 6.48, 7.51, 8.70, 10.07, 11.67, 13.51, and 15.65 for $p = 500$. The selected tuning parameters in the 20 simulation runs ranged $4.17-13.51$, which fell into the middle range of tuning parameter pool and did not cause concern in terms of variable selection or clustering performance. For $p = 1000$ the pool included numbers 2.00, 2.37, 2.82, 3.35, 3.97, 4.72, 5.60, 6.65, 7.90, 9.38, 11.14, 13.22, 15.70, 18.64, and 22.14. The selected tuning parameters in the 20 simulation runs ranged $2-15.70$. Some of these selected tuning parameters were too small: The tuning parameter 2 was chosen in four runs; 2.37 in two runs and 2.82 in one run. For these seven runs, the number of variables with nonzero weights ranged $10-19$ and the CER values had an average of 0.32 and ranged $0.27-0.40$. The tuning

parameter 15.70 was chosen in two simulation runs. The numbers of nonzero weights were 646 and 640 and the CER values were 0.30 and 0.26. For the remaining 11 runs with tuning parameter in the range of $3.97 - 7.90$, the number of variables with nonzero weights ranged $35 - 177$ and the CER values had an average of 0.19 and ranged $0.12 - 0.30$. In summary the adjusted tuning parameter pool worked fine for the scenario "$\mu = 0.6$ & $p = 500$" by shrinking the range of tuning parameter pool on both ends; But it did not work well for the scenario "$\mu = 0.6$ & $p = 1000$", where tuning parameters too small, or too large, were chosen.

3.5 THE PROPOSED METHOD: USING AN ADJUSTED BIC FOR TUNING PARAMETER CHOICE IN SPARSE K-MEANS CLUSTERING

3.5.1 The Proposed Method

We propose to use an adjusted BIC for the tuning parameter choice in the sparse k-means method [35]. It is proposed for the following reasons:

- Instead of using the distance-based approach, the model-based approach under the likelihood framework should be more sophisticated.
- The permuted datasets used for calculating the gap statistic may not represent the null distribution that it is supposed to be, because the columns with more variability will still retain that higher variability.
- The gap statistic does not account for the correlation of the variables, which we may try to incorporate by the likelihood approach.

The proposed method has the following steps:

1. Specify a tuning parameter pool with m tuning parameters.
2. For each of the tuning parameter choice s_i, where $i = 1, 2, \ldots, m$,
   a. Sparse k-means is applied to the dataset with this the tuning parameter s_i. The number of clusters K is pre-specified and stays the same.

b. Estimate the mixture distribution of K multivariate normal distributions. Each cluster is a distribution component. First, the number of variables with nonzero weight is q and the mixture distribution is described by this subset of q variables. For each cluster C_k $(k = 1, 2, \ldots, K)$, the means, μ_k, are the cluster center values and the covariance, Σ_k, are the covariance taken for that cluster. Here, $\pi_k = n_k/n$ with n_k being the number of observations in that cluster and n being the total number of observations in the dataset.

c. Compute the likelihood of datasets with the distribution parameters from last step, that is:

$$\log L = \sum_{i=1}^{n} \log[\sum_{k=1}^{K} \pi_k f_k(x_i; \mu_k, \Sigma_k)], \tag{3.10}$$

where

$$f_k(x; \mu_k, \Sigma_k) = \frac{1}{(2\pi)^{q/2} |\Sigma_k|^{\frac{1}{2}}} exp\left[-\frac{1}{2}(x_q - \mu_k)' \Sigma_k^{-1} (x_q - \mu_k)\right], \tag{3.11}$$

where x_q is the subset of the q variables.

d. Calculate the modified BIC $adBIC$ by adjusting for the different number of variables selected for clustering as the following:

$$adBIC = -2 \log L + d \log(n), \tag{3.12}$$

where $d = K - 1 + Kq + Kq(q+1)/2$

3. Choose the tuning parameter with the smallest BIC for the final sparse k-means clustering.

3.5.2 A Simple Example

We use a simple imaging dataset to illustrate this method. There are $n = 62$ subjects in the dataset and $p = 13$ regions-of-interest observed for each subject. We group the dataset into $K = 2$ groups. Table 7 shows that when the tuning parameter is 2.798409, the adjusted BIC reaches the minimum value at the first time, hence we will choose this tuning parameter for the final sparse k-means clustering. The gap statistic method also has chosen the same tuning parameter 2.798409.

Table 7: The likelihood and BIC values from an imaging dataset with regional measures with $p = 13$ regions.

Index	Tuning Parameter	Log likelihood	q	d	ad BIC
1	1.100000	44.09473	2	11	-42.79097
2	1.197452	44.09473	2	11	-42.79097
3	1.303537	43.10259	2	11	-40.80669
4	1.419020	121.83404	3	19	-165.25253
5	1.544735	121.83404	3	19	-165.25253
6	1.681587	273.32955	5	41	-377.44658
7	1.830563	351.46995	6	55	-475.94752
8	1.992737	351.46995	6	55	-475.94752
9	2.169278	351.46995	6	55	-475.94752
10	2.361460	590.90537	9	109	-731.95309
11	2.570668	807.67813	12	181	-868.34493
12	2.798409	1041.36069	13	209	-1220.15030
13	3.046327	1041.36069	13	209	-1220.15030
14	3.316209	1041.36069	13	209	-1220.15030
15	3.610000	1041.36069	13	209	-1220.15030

3.5.3 Simulation Results

We are applying the proposed method to the simulation scenarios where the gap statistic has failed. We are experiencing some issues and in the middle of resolving them. The following problems can occur when calculating the multivariate normal distribution density:

- Sometimes the covariance matrix is singular and we cannot calculate its inverse. To get around this problem, we make the covariance matrix positive definite.
- The determinant for the covariance matrix is nearly zero, which in turn gave an infinite density value and hence an infinite likelihood value.

These problems exist not only for cases where $p = 1000$ and $p = 500$, but also when $p = 200$ where the gap statistic worked well.

3.6 CONCLUSIONS AND DISCUSSIONS

The tuning parameter choice for sparse clustering is inherently difficult and challenging because of the lack of information on the truth. However, it is also very important given that sparse clustering is becoming widely applied in the high dimensional setting. The implementation of our proposed method is difficult because of the problems with the covariance matrix. Looking closely to work done by the Pan and Shen[2007] paper, we can see that they assume a diagonal covariance matrix and that the density of the multivariate distribution is calculated as the product of the individual one-dimensional normal distribution densities to get around the problems that we are experiencing. Another challenge using this proposed method is that the estimated parameters, such as the covariance matrix, may be biased from the true values in the high-dimensional setting. There are shrinkage methods in the literature where shrinking the estimated covariance is desired to get close to the true variance.

Another idea is to use the prediction strength for tuning parameter choice as in [32]. This idea is similar to cross-validation. A simple approach is to split the data into two datasets. Sparse clustering will be fitted to one dataset with each tuning parameter choice. We will

then check the prediction strength of the cluster criteria in the other half of the data by using the metric Classification Error Rate (CER), which is equal to "1 - Rand index". The tuning parameter with maximum prediction strength will be chosen to be the final clustering model. We had reservations about exploring this approach for two reasons: one reason is that when the sample size is limited, cross-validation is less valid; Another reason is that the prediction strength of the clustering again is difficult to estimate because of the lack of the information on the true group membership.

In summary, the problems that we experienced with the gap statistic alerted us that one should exercise caution when applying the method. More work should be done to improve the situation.

4.0 OTHER WORK

Besides the work described in previous chapters, the resampling method described in Chapter 2 has also been extensively applied in imaging datasets with regional values ($p = 13$). The results were included in a paper where the goal was to find the amyloid positivity threshold for Alzheimer's disease early detection. In this work, we resampled each dataset for 1000 times, defined the robust cluster centers as the 95% tight cluster centers and then we use these robust cluster centers to further define robust clustering rules and amyloid positivity threshold. This paper has been submitted to Journal of Nuclear Medicine for peer review.

BIBLIOGRAPHY

[1] Jeffrey D. Banfield and Adrian E. Raftery. Model-based gaussian and non-gaussian clustering. *Biometrics*, 49(3):803–821, 1993.

[2] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, United Kingdom, 2004.

[3] Emmanuel Candes and Terence Tao. The dantzig selector: statistical estimation when p is much larger than n. *Annals of Statistics*, 35:2313–2351, 2007.

[4] Hyonho Chun and Sndz Keles. Sparse partial least squares regression for simultaneous dimension reduction and variable selection. *Journal of the Royal Statistical Society, Series B*, 72(1):3–25, 2010.

[5] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *Annals of Statistics*, 32(2):407–499, 2004.

[6] E.W. Forgy. Cluster analysis of multivariate data: efficiency vs interpretability of classifications. *Biometrics*, 21:768–769, 1965.

[7] Chris Fraley and Adrian E. Raftery. How many clusters? which clustering methods? answers via model-based cluster analysis. *The Computer Journal*, 41(8):578–588, 1998.

[8] Chris Fraley and Adrian E. Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458):611–631, 2002.

[9] Jerome Friedman, Trevor Hastie, Holger Hoefling, and Robert Tibshirani. Pathwise coordinate optimization. *Annals of Applied Statistics*, 1(2):302–332, 2007.

[10] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.

[11] Wenjiang J. Fu. Penalized regressions: The bridge versus the lasso. *Journal of Computational and Graphical Statistics*, 7(3):397–416, 1998.

[12] Alexander Genkin, David D. Lewis, and David Madigan. Large-scale bayesian logistic regression for text categorization. *Technometrics*, 49(3):291–304, 2007.

[13] John. A. Hartigan and M. A. Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

[14] Trevor Hastie, Saharon Rosset, Robert Tibshirani, and Ji Zhu. The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, 5:1391–1415, 2004.

[15] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York, 2009.

[16] Jr. Joe H. Ward. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963.

[17] Ian T. Jolliffe, Nickolay T. Trendafilov, and Mudassir Uddin. A modified principal component technique based on the lasso. *Journal of Computational and Graphical Statistics*, 12(3):531–547, 2003.

[18] Leonard Kaufman and Peter J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley, New York, 1990.

[19] William E. Klunk, Henry Engler, Agneta Nordberg, Brian J. Bacskai, Yanming Wang, Julie C. Price, Mats Bergstrom, Bradley T. Hyman, Bengt Langstrom, and Chester A. Mathis. Imaging the pathology of alzheimer's disease: Amyloid-imaging with positron emission tomography. *Neuroimaging Clinics of North America*, 13(4):781–9, ix, 2003.

[20] Stuart P. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28:129–137, 1982.

[21] James B. MacQueen. Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, eds L. M. Le Cam and J. Neyman*, pages 281–297, 1967.

[22] Chester. A. Mathis, Yanming Wang, Daniel P. Holt, Guo-Feng Huang, Manik L. Debnath, and William E. Klunk. Synthesis and evaluation of 11c-labeled 6-substituted 2-arylbenzothiazoles as amyloid imaging agents. *Journal of Medicinal Chemistry*, 46(13):2740–54, 2003.

[23] Geoffrey J. McLachlan, Richard W. Bean, and David Peel. A mixture model-based approachto the clustering of microarray expression data. *Bioinformatics*, 18(3):412–422, 2002.

[24] Geoffrey J. McLachlan and David Peel. *Finite Mixture Model*. John Wiley & Sons, Inc, New York, 2002.

[25] Geoffroy J. McLachlan and Kaye E. Basford. *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, New York, 1988.

[26] Lukas Meier, Sara van de Geer, and Peter Bühlmann. The group lasso for logistic regression. *Journal of the Royal Statistical Society, Series B*, 70(1):53–71, 2008.

[27] Wei Pan and Xiaotong Shen. Penalized model-based clustering with application to variable selection. *Journal of Machine Learning Reseach*, 8:1145–1164, 2007.

[28] Gideon Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6(2):461–464, 1978.

[29] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58(1):267–288, 1996.

[30] Robert Tibshirani. The lasso method for variable selection in the cox model. *Statistics in Medicine*, 16:385–395, 1997.

[31] Robert Tibshirani. Regression shrinkage and selection via the lasso: a retrospective. *Journal of the Royal Statistical Society, Series B*, 73(3):273–282, 2011.

[32] Robert Tibshirani and Guenther Walther. Cluster validation by prediction strength. *Journal of Computational and Graphical Statistics*, 14(3):511–528, 2005.

[33] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society, Series B*, 63(2):411–423, 2001.

[34] George C. Tseng and Wing H. Wong. Tight clustering: a resampling-based approach for identifying stable and tight patterns in data. *Biometrics*, 61(1):10–16, 2005.

[35] Daniela M. Witten and Robert Tibshirani. A framework for feature selection in clustering. *Journal of the American Statistical Association*, 105(490):713–726, 2010.

[36] Daniela M. Witten, Robert Tibshirani, and Trevor Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10(3):515–534, 2009.

[37] Tong Tong Wu and Kenneth Lange. Coordinate descent procedures for lasso penalized regression. *The Annals of Applied Statistics*, 2(1):224–244, 2008.

[38] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68(1):49–67, 2006.

[39] Ming Yuan and Yi Lin. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.

[40] Hui Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2005.

[41] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67(2):301–320, 2005.